Attorney Docket No: 53296-2

DECLARATION AND POWER OF ATTORNEY

As the below-named inventor, I declare that I am the original, first, and sole inventor of the subject matter which is claimed in the patent application identified below and for which a patent is sought on the invention as titled therein. I hereby state that I have reviewed and understood the contents of said specification, including the claims. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with 37 C.F.R. § 1.56(a).

Inventor: Sajid Ahmed

Address: P.O. Box 153
La Cañada, CA 91012

Residence: La Cañada, California, USA

Citizenship: USA

Title of Invention: **SYSTEM AND METHOD FOR DECISION MAKING**

Serial No.: to be assigned, filed on January 8, 2001

[ ] There are no earlier-filed United States Patent Applications of which priority benefit is claimed.

[X] I hereby claim the benefit under 35 U.S.C. § 120 of the United States Patent Application listed below, and, insofar as the subject matter of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose material information as defined in 37 C.F.R. § 1.56(a) which occurred between the filing date of the prior application and the filing date of this application:

USSN:	Filed:	Status:
60/175,106	06 January 2000	pending

POWER OF ATTORNEY

The power to prosecute this application and transact all business in the Patent and Trademark Office connected herewith is hereby granted to the following attorneys:

Todd M. Becker, Registration No. 43,487
Vita Conforti, Registration No. 39,639
Barry L. Davison, Registration No. 47,309

Please send all correspondence to:

Barry L. Davison
Davis Wright Tremaine LLP
2600 Century Square
1501 Fourth Avenue
Seattle, WA 98101-1688
Telephone: (206) 628-7621
Facsimile: (206) 628-7699

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor: Sajid Ahmed

Sajid Ahmed

Date signed: 1-8-2001

Appendix A

```
#!/usr/bin/perl -w

$html_title = $0;
$html_title =~ s/.*\/(\w+)$/$1/;
$cgi_name = $html_title;
$html_title =~ s/_/ /g;

$[ = 1;
$debug = 0;
$answercookiename = "chaincookie";
$answercookiefilter = "[-0123456789:.]+";
$levelcookiename = "chainlevel";
@questions_2b_asked = ();

($num_questions,$num_conditions,$max_level) = read_dat_file();
if ($debug) { addbody("# questions=$num_questions, # conditions=$num_conditions, max
lvl=$max_level<br>\n"); }

($first_unanswered_question,$level) = get_state_from_cookie($num_questions);
if ($debug) { addbody("get_state_from_cookie:first=$first_unanswered_question,level=$level<br>\n");}

update_state_from_form();

$new_answercookie = join(":", @answers);
$new_levelcookie = $level+1;
if ( $debug ) {
  addbody("new_level=$new_levelcookie\n<br>");
  addbody("new answer cookie = $new_answercookie\n");
  if ( $new_answercookie =~ /$answercookiefilter/ ) {
    addbody("new_answercookie passed filter\n");
  }
}

if ($debug) { addbody("\n\n\n"); }

if ($level > $max_level) {
  while (($diag, $factor_list) = each %condition) {
    if ( $debug ) { addbody("$diag (has factors $factor_list)<br>\n"); }
    @factors = split(',', $factor_list);
    if ( $debug ) { addbody("$#factors items in \@factors<br>\n"); }
    $score = 0; $max = 0; $num_answers=0;
    foreach $factor (@factors) {
      ($question,$weight) = $factor =~ /q(\d+)=(\d+\.*\d*)/;
      $max = $max + $weight;
      if ( $answers[$question] > 0 ) {
        $score = $score + ( $weight * $answers[$question] );
        $num_answers++;
      }
      $answer = $answers[$question];
      if ( $debug ) { addbody("  pondering $question $weight.  It was \"$answer\" so new score =
$score<br>\n"); }
      $question++;
    }
```

```
    if ( $num_answers==0 ) { $num_answers=1; }
    $score = int( $score / $num_answers * 100) / 100;
    if ( $debug ) { addbody( "   final score = $score ($max max)\n"); }

    $a{$diag} = $score;
    $question++;
  }
  if ( $debug ) { print "\n"; }
  @unsorted = ();
  foreach $diag (keys(%a)) {
    $url = $diag;
    $url =~ s/[ \/]/_/g;
    $url =~ s/["]//g;
    $url = "http://adsl-63-194-251-2.dsl.lsan03.pacbell.net/igotpain/$cgi_name/$url.html";
    @unsorted = ( "$a{$diag}\t\t<a href=\"$url\" TARGET=\"reference\">$diag</a>\n", @unsorted );
  }
  if ( $debug ) { addbody("</PRE>\n"); }
  addbody("<PRE>\n");
  addbody( "% Possibility\tCondition\n");
  addbody(reverse(sort sortdiags @unsorted));
  addbody("</PRE>\n");
  if ( $debug ) { addbody("</PRE>\n"); }
  addbody("(Back to <A HREF=\"/igotpain/welcome.html\" TARGET=\"reference\">vitruvian man</A>)");
  if ( $debug ) { addbody("<PRE>\n"); }
  $new_answercookie = "";
  $new_levelcookie = "";
}
 else { # ask another set of questions
  if ($debug) { addbody("</PRE>\n"); }
  addbody("<FORM method=\"POST\" action=\"/cgi-bin/igp/$cgi_name\">\n");
  @questions_2b_asked = build_question_list($level, $max_level, $first_unanswered_question,
$num_questions);
  $new_answercookie = join(":", @answers);
  $new_levelcookie = $level+1;
  if ( $debug ) {
    addbody("new_level=$new_levelcookie, old_level=$level\n<br>");
    addbody("new answer cookie = $new_answercookie\n");
    if ( $new_answercookie =~ /($answercookiefilter)/ ) {
      addbody("new_answercookie passed filter\n");
    }
  }
  $current_header = "";
  foreach $qdata (@questions_2b_asked) {
    ($qnum,$level,$depend,$help_ref,$question,$qheader) = split("xyzzy",$qdata);
    if ( $current_header ne $qheader ) {
      $current_header = $qheader;
      addbody("<TABLE BGCOLOR=\"#A0A0A0\"><TR><TD ALIGN=\"LEFT\"
COLSPAN=2><H2>$current_header</H2></TD></TR>\n");
    }
    addbody("<TR>\n");
    addbody(get_question_html("q$qnum",$question,$help_ref,"no"));
    addbody("</TR>\n");
  }
  addbody("</TABLE>\n");
  addbody("<input type=\"submit\" value=\"Proceed\">\n");
  addbody("<input type=\"reset\">\n");
```

```
  addbody("</FORM>\n");
  if ($debug) { addbody("<PRE>\n"); }
}

sub sortdiags {
  ($aa) = split("\t",$a);
  ($bb) = split("\t",$b);
  #$aa = sprintf("%3.4d",$aa);
  #$bb = sprintf("%3.4d",$bb);
  ($aa<=>$bb)
}

#############
###
### do the printout
###
#############

&printheader($new_answercookie,$new_levelcookie);
printhtml();
exit 0;

#XXXXXXXXX
sub build_question_list {
  local ($last_lev, $max_lev, $first_q, $last_q) = @_;
  local ($qlevel,$qdepend,$dep_qnum,$dep_answer);

  @qlist = ();

  if ($debug) { addbody("in build_question_list(last=$last_lev, max=$max_lev, first=$first_q,
last=$last_q)\n<br>"); }
  foreach $qnum ( $first_q..$last_q ) {
    if ( $answers[$qnum] != -1 ) {
      if ($debug) { addbody("q#$qnum skipped<br>\n");}
      next;
    }
    ($iqnum,$qlevel,$qdepend) = split("xyzzy",$q[$qnum]);
    if ($debug) { addbody("qnum = $iqnum($qnum); qlevel = $qlevel; qdepend = $qdepend<br>\n"); }
    ($dep_qnum,$dep_answer) = $qdepend =~ /(\d+)([+-]*)/;
    if ( $dep_answer =~ /\+/ ) { $dep_answer = "1"; }
    if ( $dep_answer =~ /-/ ) { $dep_answer = "0"; }
    if ($debug) { addbody("answers[$dep_qnum]=$answers[$dep_qnum], dep_qnum=$dep_qnum,
dep_answer=$dep_answer<BR>\n"); }
    if ( $qlevel == $last_lev ) {
      if ( $dep_qnum == 0 || $answers[$dep_qnum] == $dep_answer ) {
        push(@qlist,$q[$qnum]);
        if ($debug) { addbody("pushed $q[$qnum]\n"); }
      }
    }
  }
  if ( $debug ) {
    addbody( "build_question_list = \n");
    foreach $line (@qlist) { addbody( "$line\n"); }
  }
  @qlist;
}
```

```
sub update_state_from_form {
  $n = <>;
  $n = "" unless ($n);
  if ( $debug ) { addbody( "STDIN: $n\n<br>"); }
  @question_answer = split(/&/, $n);
  foreach ( @question_answer ) {
    if ( $debug ) { addbody("$_ : "); }
    if ( /q(\d+)=(\w+)/ ) {
      ($qnum,$val) = ($1,$2);
      if ( $debug ) { addbody("$qnum => $val\n"); }
      if ($val eq "yes") { $answers[$qnum] = 1; }
      if ($val eq "sometimes") { $answers[$qnum] = .75; }
      if ($val eq "maybe") { $answers[$qnum] = .45; }
      if ($val eq "unknown") { $answers[$qnum] = .2; }
      if ($val eq "no") { $answers[$qnum] = 0; }
    }
    if ( $n =~ /R=yes/ ) { $answers[$qnum] = -1; }
  }
}

sub read_dat_file {
  local ($number_of_questions,$number_of_conditions,$max_lev) = (0,0,0);
  local ($line,$condition_name,$factors) = ("","","");
  local ($qnum,$qlevel,$depend,$help_ref,$question,$current_header) = (0,0,"","","","");

  open(DAT, "<$cgi_name.dat") || die("test.dat unreadable\n");
  @dat = <DAT>;
  close(DAT);

  foreach $line (@dat) {
    if ( $line =~ /^"([^"]+)",("\d+","\d+",.*)$/ ) {
      if ( $debug ) { addbody("\n<br>condition line: $line"); }
      ($condition_name , $line) = ($1,$2);
      $number_of_conditions++;
      $line =~ s/"//g;
      @factors = split(/,/,$line);
      foreach $n (1..$#factors) { $factors[$n] = "q$n=$factors[$n]"; }
      $factors = join(",",@factors);
      $condition{$condition_name} = $factors;
      if ( $debug ) { addbody("\%condition $condition_name = $factors"); }
    }
    elsif ( $line =~ /^(\d+) (\d+) (\d+[+-]*) (\w+) (.*)/ ) {
      if ( $debug ) { addbody("question line: $line"); }
      ($qnum,$qlevel,$depend,$help_ref,$question) = ($1,$2,$3,$4,$5);
      $q[$qnum] = join("xyzzy",($qnum,$qlevel,$depend,$help_ref,$question,$current_header));
      $number_of_questions++;
      if ( $qlevel > $max_lev ) { $max_lev = $qlevel; }
    }
    elsif ( $line =~ /^H (.*)/ ) { $current_header = $1; }
    else {
      if ( $debug ) { addbody("unprocessed line:\n$line\n"); }
    }
  }
  if ( $debug ) {
```

```
    addbody("number_of_questions = $number_of_questions, number_of_conditions =
$number_of_conditions, max_lev = $max_lev\n");
  }
  ($number_of_questions,$number_of_conditions,$max_lev);
}


### Cookies ####

sub get_state_from_cookie {
  local ($qnum) = @_;
  local ($first,$qlevel) = (1,1);

  $env_cookie = $ENV{HTTP_COOKIE};
  if ( $debug) {
    addbody("env_cookie = $env_cookie \n");
    addbody("cookiefilter = $answercookiefilter \n");
  }
  if ( $env_cookie =~ /$answercookiename=($answercookiefilter)/ ) { $cookie = $1; }
  if ( $env_cookie =~ /$levelcookiename=(\d+)/ ) { $qlevel = $1; }
  if ( $debug ) { addbody("filtered cookie = $cookie\n"); }
  if ( $cookie ) {
    @answers = split(/:/, $cookie);
    until ( ($answers[$first] == -1) || ($first > 80) ) { $first++; }
  }
   else {
    foreach $n (1..$qnum) { $answers[$n] = -1; }
    $first = 1;
    if ( $debug ) { addbody("initialized $qnum answers to -1\n"); }
   }
  ($first,$qlevel);
}


### HTML stuff ###

sub get_question_html {
  local ($name,$question,$help,$default) = @_;
  local ($html, $yeschecked, $nochecked) = ("","","");

  if ( $default eq "yes" ) { $yeschecked = "SELECTED"; }
  if ( $default eq "no" ) { $nochecked = "SELECTED"; }

  $html = "\
  <TD ALIGN=\"CENTER\">\
  <SELECT NAME=\"$name\">\
   <OPTION $yeschecked VALUE=\"yes\">Yes\
   <OPTION VALUE=\"sometimes\">Sometimes\
   <OPTION VALUE=\"maybe\">Maybe\
   <OPTION VALUE=\"unknown\">Don\'t remember\
   <OPTION $nochecked VALUE=\"no\">No\
  </SELECT></TD>\
  <TD>$question<a href=\"/igotpain/$cgi_name/$help.html\" TARGET=\"reference\">(help)</a></TD>\
  \n";

  return($html);
```

```
}

sub addbody {
  push(@body_lines, @_);
}

sub printtitle {
  local ($title) = @_;

  print "<TITLE>$title</TITLE>";
}

sub printbody {
  print "<BODY>";
  if ($debug) {print "<PRE>";}
  print @body_lines;
  if ($debug) {print "</PRE>";}
  print "</BODY>";
}

sub printhtml {
  print "<HTML>\n";
  &printtitle("$html_title");
  &printbody;
  print "</HTML>\n";
}

sub printheader {
  local ($cookie,$qlevel) = @_;

  print "Content-type: text/html\n";
  print "Set-Cookie: $answercookiename=$cookie\n";
  print "Set-Cookie: $levelcookiename=$qlevel\n";
  print "\n";
}
```